

## i Om eksamen

Eksamensinformasjon – digital skoleeksamen

Fakultet: Teknologi, kunst og design

Utdanning: Teknologiske fag

Emnenavn: Algoritmer og Datastrukturer

Emnekode: (KONT) DATS2300 / ITPE2300

---

Dato: 28.02.2019

Tid: 09.00 - 12.00

Antall oppgaver: 5

Tillatte hjelpemidler: Ingen

Merknad:

Råd og tips:

1. Les gjennom hele oppgavesettet før du begynner og planlegg tiden.
2. Svar utfyllende på oppgavene så du viser at du har forstått pensum.
3. Bruk ikke for lang tid på et punkt. Gå isteden videre til neste punkt og eventuelt tilbake hvis du får god tid.
4. Hvis du trenger en hjelpestruktur (liste, stakk, kø o.l.) fra java.util eller fra kompendiet, kan du fritt bruke den uten å måtte kode den selv. Men den må brukes på en korrekt måte. Men du bør si fra om dette i en kommentar.
5. Hvis du har idéer om hvordan ting skal løses, men likevel ikke klarer å få det til, kan du demonstrere idéene dine med ord, tegninger o.l.

Vurdering:

1. Ved sensurering blir det lagt vekt på hva du viser av forståelse av kursets pensum opp mot læringsutbyttet. Det vil si at det vektlegges hvordan du kommer frem til et svar ved å bruke pensum.
2. De fem oppgavene teller likt.

<sup>1</sup> **Oppgave 1: Iterasjon, rekursjon, og lenkede lister**

**I denne oppgaven handler om rekursjon og lenkede lister**

- a. Forklar kort hva forskjellen er på en iterativ og en rekursiv funksjon.
- b. Skriv en programkode i Java som skriver ut verdien av nodene i et den lenkede listen gitt i vedlegget. Bruk kildekoden i vedlegget og fyll ut innholdet i funksjonen `printIterative()`
- c. Skriv en programkode i Java som skriver ut verdien av nodene i et den lenkede listen gitt i vedlegget. Bruk kildekoden i vedlegget og fyll ut innholdet i funksjonen `printRecursive(Node node)`
- d. Hva vil utskriften av programmet i vedlegget bli?

**Skriv ditt svar her...**

Maks poeng: 20

2

Oppgave 2: Huffmantrær

I denne oppgaven skal du bruke et Huffmantre til å kode en tekst.

- a. Lag det venstreorienterte kanoniske Huffmantreet ut ifra følgende bokstavfrekvenser:  
A: 10, B: 12, C: 14, D: 16, R: 18.

i. Lag en tegning av Huffmantreet, og

ii. Skriv opp en tabell med Huffmankoder for hver bokstav
- b. Bruk så Huffmankodene til å komprimere «ABRA».

i. Skriv opp den kodede binære meldingen.

ii. Hvor mange bit bruker du i den kodede meldingen?

Skriv ditt svar her...

Maks poeng: 20

3

## Oppgave 3: Sortering

I denne oppgaven skal bruke flettesortering (merge sort) og kvikksortering (quicksort) til å sortere en tallrekke.

- a. Ta utgangspunkt i tallrekken 9, 8, 7, 6, 5, 4, 3, 2, 1. Lag en tegning som illustrerer hvordan flettesortering (merge sort) sorterer tallene i stigende rekkefølge, og beskriv kort med tekst hva hovedprinsippet er.
- b. Ta utgangspunkt i samme tallrekke som i deloppgave a. Lag en tegning som illustrerer hvordan kvikksortering (quicksort) sorterer tallene i stigende rekkefølge, og beskriv kort med tekst hva hovedprinsippet er.
- c. Forklar hva begrepene verste tilfellet (worst case) og beste tilfellet (best case) betyr når vi snakker om en algoritmes kompleksitet / effektivitet.

Skriv ditt svar her...

Maks poeng: 20

4    **Oppgave 4: Dijkstras algoritme**

I denne oppgaven skal vi finne en vei i en graf ved hjelp av Dijkstras algoritme. I denne oppgavene er det viktig at du viser at du kan Dijkstras algoritme.

- a. Beskriv med ord hva Dijkstras algoritme er.
  - i. Hva bruker du Dijkstras algoritme til?
  - ii. Hvordan fungerer Dijkstras algoritme?
  - iii. Hvilken hjelpedatastruktur benyttes i Dijkstras algoritme?
- b. Ta utgangspunkt i grafen i vedlegget. Lengden på hver kant står i firkantene.
  - i. Lag en tegning som viser hvordan Dijkstras algoritme finner en vei fra F til H.
  - ii. Hvilke noder passerer du og hva er lengden av veien?

**Skriv ditt svar her...**

Maks poeng: 20

5    **Oppgave 5: 2-3-4 trær (B-tre av orden 4)**

I denne oppgaven skal du tegne forskjellige binære søketrær.

- a. Start med et tomt binært søketre og legg inn tallene 15, 4, 11 (i den gitte rekkefølgen). Tegn treet.
- b. Fortsett å legge inn tallene 9, 16, 29, 6 i treet. Tegn treet etter å ha lagt inn disse tallene.
- c. Du skal nå legge inn de samme tallene i et 2-3-4 tre.
  - i. Start med ett tomt tre og tegn treet etter å ha lagt inn 15, 4, 11.
  - ii. Legg så til 9, 16, 29, 6, og tegn treet for hvert tall du legger til.
- d. Hva er fordelene med å bruke et 2-3-4 tre i denne oppgaven?

Skriv ditt svar her...

Maks poeng: 20

## Question 1

Attached



```

/**
 * Klasse som representerer en node i en enkelt lenket liste
 * next er neste node i listen
 * value er verdien denne noden holder
 */
public static class LinkedList {

    /**
     * En node i den lenkede listen
     */
    public static class Node {
        Node next;
        char value;

        /**
         * Konstruktør som lager en ny node
         * @param value Verdien den nede noden skal ha
         */
        Node(char value) {
            this.value = value;
        }
    }

    //Peker til hode (første element i listen)
    Node head;

    /**
     * Konstruktør
     */
    LinkedList() {
        head = null;
    }

    /**
     * Funksjon som legger til en verdi i den lenkede listen
     * @param value
     */
    void add(char value) {
        Node new_node = new Node(value);
        new_node.next = head;
        head = new_node;
    }
}

```



```

/**
 * Funksjon som skriver ut ved å iterere gjennom listen
 */
public void printIterative() {
    //Her skal din kode komme i oppgave c
}

/**
 * Funksjon som skriver ut ved å bruke rekursjon
 */
public void printRecursive() {
    //Her skal din kode komme i oppgave d (del 1 av 2)
}

/**
 * Hjelpemetode for den rekursive utskriftsfunksjonen
 * @param node
 */
private void printRecursive(Node node) {
    //Her skal din kode komme i oppgave d (del 2 av 2)
}
}

public static void main(String[] args) {
    LinkedList liste = new LinkedList();
    char values[] = "ABCDEFGHIJ".toCharArray();

    for (char val : values) {
        liste.add(val);
    }

    System.out.println("Rekursiv utskrift: ");
    liste.printRecursive();

    System.out.println("Iterativ utskrift: ");
    liste.printIterative();
}

```

## Question 4

Attached



